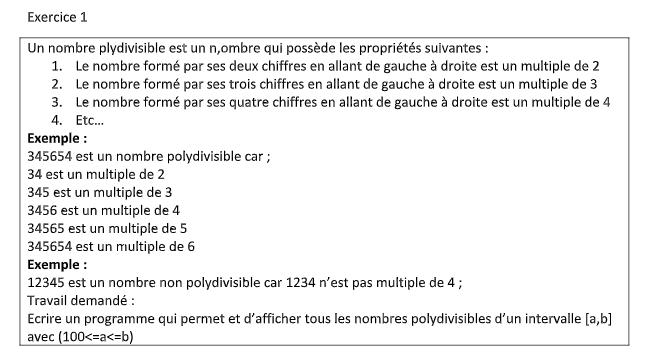
**Application :**



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Algorithme plydivisible :***   |  |  | | --- | --- | | TDO globaux | | | a, b | Entier | | Affiche | Procédure | | remplir | Fonction |   Début  a, b <- remplir ()  affiche (a, b)  Fin  Fonction remplir ()  Début   |  |  | | --- | --- | | TDO locaux | | | Objet | Type | | a, b | entier |   répéter  Lire(a)  Lire(b)  Jusqu’à (100 <= a <= b)  Retourner a , b  Fin  Fonction polydivisible (x : entier) : booléen  i <- 1  sa <- convch (x)  ply <- Vrai  Tant que (i <= long(sa)) alors  T = sous chaine (sa,0,i)  Si int(t) mod long(t) ≠ 0 alors  Ply <- Faux   |  |  | | --- | --- | | TDO locaux | | | i | Entier | | Sa, t | Chaine de caractère | | ply | Booléen |   i <- i + 1  Retourner ply  Fin  Procédure Affiche (a,b : entier)  Pour i de a à b faire  Si polydivisible (i) :  Ecrire (i)   |  |  | | --- | --- | | TDO locaux | | | Objet | Type | | polydivisible | fonction | | i | entier |   Fin | ***Python :***  def remplir () :  a = int(input("donner un entier a : "))  b = int(input("donner un entier b : "))  while not (100 <= a <= b) :  a = int (input("donner un entier a : "))  b = int(input("donner un entier b : "))  return a , b  def polydivisible (x) :  i = 1  sa = str(x)  ply = True  while i <= len(sa) :  t = sa[0:i]  if int(t) % len(t) != 0 :  ply = False  i = i + 1  return ply  def affiche (a , b) :  for i in range (a , b) :  if polydivisible(i) == True :  print(i)  return  a , b= remplir()  affiche(a , b) |